Introduction au calcul scientifique avec Matlab

TP3

Correction

Exercice 1.

Observer l'effet des instructions suivantes.

>>x= -pi :0.1:3*pi; => crée un vecteur de –pi à 3pi avec un pas de 0.1

>>y=x.*sin(x); => retourne le vecteur résultat. y a la même taille que x

>>plot(x,y) => ouvre une fenêtre contenant la figure qui trace les couples des valeurs (x,y)

>>clf => nettoie la fenêtre de la figure en la supprimant

>>plot(x,y); axis([-pi,3*pi,-6,9]); => précise les valeurs max et min des axes x et y (axis([XMIN XMAX YMIN YMAX]))

>>xlabel('x'); ylabel('y'); => Nome l'axe x 'x', et l'axe y 'y'

>>title(['graphe de la fonction x sin(x) sur l''intervalle [' num2str(x(1)) ' , ' num2str(x(end)) ']']) => inscrit le titre 'graphe de la fonction x sin(x) sur l''intervalle [-3.1416 , 9.3584]' au dessus de la figure

>>plot(x,y,x,2*y) => trace, sur le même graphique, les couples des valeurs (x,y) et (x,2*y)

>>plot(x,[y;2*y]) => fait la même chose que l'instruction plot(x,y,x,2*y)

>>plot(x,y,'r--',x,2*y,'g+') => trace la courbe (x,y) avec un traie discontinu rouge ('r--'), et la courbe (x, 2*y) avec les marques + en vert ('g+')

>>fplot('[x*sin(x),2*x*sin(x)]',[-pi,3*pi]) => trace, sur l'intervalle [-pi,3*pi], les courbes des deux fonctions x*sin(x) et 2*x*sin(x)

>>fplot('x*sin(x)',[-pi,3*pi],'b-') hold on => trace, sur l'intervalle [-pi,3*pi], la courbe de la fonction x*sin(x) avec un traie bleu continu ('b-'). L'instruction hold on mais la figure en attente pour y tracer une autre courbe.

>>fplot('2*x*sin(x)',[-pi,3*pi],'yo') hold off => trace, sur la même figure précédente, la courbe de la fonction 2*x*sin(x) avec les marques o en jaune ('yo')

>>t=0:0.1:2*pi; => crée un vecteur de o à 2pi avec un pas de 0.1

>>plot(sin(t),sin(2*t)) => trace les couples des valeurs (sin(t),sin(2*t)) pour l'ensemble des valeurs du vecteur t préalablement créé

>>plot(sin(t),sin(2*t),'c-') => trace les couples des valeurs (sin(t),sin(2*t)) avec un traie continu de couleur turquoise

Exercice 2.

Soit f et g les fonctions définies sur l'intervalle [0, 10] par :

$$f(x) = \exp\left(-\frac{x}{25}\right), \quad g(x) = \cos\left(\frac{x}{10}\right)$$

Tracez ces deux fonctions (à l'aide de fplot) d'abord dans une même fenêtre graphique mais sur des graphes différents, puis dans une même fenêtre graphique et sur le même graphe.

Réponse :

```
>> fplot('exp(-z/25)',[0,10]) hold on
>> fplot('cos(s/10)',[0,10]) hold off

>> fplot('[exp(-z/25),cos(z/10)]',[0,10])
```

Exercice 3.

1. Tirer 100 couples de points (x, y) aléatoirement dans le carré [0, 1] × [0, 1].
2. Représenter le nuage de points obtenus dans une fenêtre graphique.
3. Calculer le centre de gravité G du nuage de points.
4. Ajouter en rouge au nuage de points le centre de gravité.
5. Sauvegarder la figure sous le nom nuage.fig. Fermer la fenêtre graphique. Ouvrir une nouvelle fenêtre graphique et faire réapparaître le fichier nuage.fig.

Réponse :

1. >> x=rand(1,100) ;
   >> y=rand(1,100) ;

2. >> plot(x,y) hold on

3. Doit g le centre de gravité du nuage de point (x,y) :
>> xg = mean(x) ;
>> yg = mean(y) ;

4. >> plot(xg,yg, 'ro') => rajoute, sur le graphique précédent, le point g avec une marque o rouge

5. >> savefig('nuage.fig') => sauvegarde dans le dossier courant, qui est visible en haut à gauche de la fenêtre Matlab, un fichier nuage.fig

>> openfig('nuage.fig') => ouvre la figure préalablement sauvegardée sous le nom nuage.fig

Exercice 4.

Reprendre l'exercice précédent et faire afficher dans une même fenêtre graphique deux histogrammes, un pour les abscisses et l'autre pour les ordonnées des points tirés.

N'oublier pas la commande help pour savoir comment tracer un histogramme !

Réponse :

>> histogram(x) hold on
>> histogram(y)

Exercice 5.

Soit la matrice et les vecteurs colonnes suivants :

$$A = \begin{pmatrix} 5/8 & -1/4 & 1/8 \\ 1/4 & 0 & 1/4 \\ 1/8 & -1/4 & 5/8 \end{pmatrix}, \vec{b} = \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}, \vec{u}_1 = \begin{pmatrix} 5 \\ 2 \\ -4 \end{pmatrix}$$

On définit, pour n ≥ 1, la suite de vecteurs $u_{n+1} = Au_n + b$.

1. Construire une fonction suite.m calculant les premiers termes de la suite u_n. Cette fonction aura comme arguments d'entrée les données suivantes : la matrice A, le second membre b, le terme initial u_1, et le nombre de termes voulus nb.
2. Représenter graphiquement l'évolution de chacune des composantes. Qu'observe-t-on ?
3. Soient

$$\vec{u}_{1b} = \begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix}, A_b = \begin{pmatrix} 5 & 6 & 3 \\ -1 & 5 & -1 \\ 1 & 2 & 0 \end{pmatrix}$$

Observe-t-on le même comportement si on remplace u_1 par u_{1b} ? Que se passe-t-il si on remplace A par A_b (quel que soit le terme initial) ?

Réponse :

1. - Créer un fichier suite.m
   - Y mettre le code suivant :

```
function u = suite(A,u1,b,nb)
%SUITE calcule les nb_it premiers termes de la suite Un = A*Un + b
 % En sortie, le tableau u contient les itérés successifs (Un)1<=n<= nb
u = zeros(3,nb+1);
u(:,1) = u1;
for k = 1:nb
u(:,k+1) = A*u(:,k)+b;
end
end
```

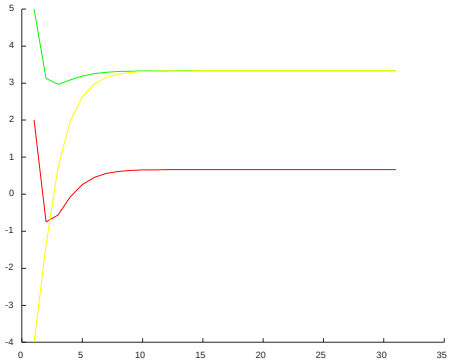
2. Pour représenter graphiquement de chacune des composante de la suite, on tape d'abord :

```
>> A = [ 5/8 -1/4 1/8 ; 1/4 0 1/4 ; 1/8 -1/4 5/8 ]
>> b = [ 1 ; -1 ; 1 ]
>> u1 = [ 5 ; 2 ; -4 ]
```

Puis on calcule, en utilisant la fonction suite.m, par exemple les 30 premiers termes de la suite u_n
>>u = suite(A,u1,b,30);

Enfin, on trace sur la même figure les trois composantes des 30 termes calculés, en prenant le soin d'utiliser des couleurs différentes

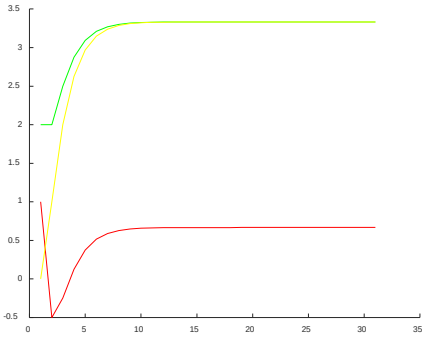
>>hold on
>>plot(u(1,:),'g')
>>plot(u(2,:),'r')
>>plot(u(3,:),'y')
>>hold off



Les itérations convergent visuellement vers (10/3, 2/3, 10/3).

3. Remplaçant u_1 par u_{1b}, et calculant les 30 premiers termes de la suite pour observer la convergence de la suite :

>> u1b = [2 ; 1 ; 0];
>> ub = suite(A,u1b,b,30);
>> hold on
>> plot(ub(1,:),'g')
>> plot(ub(2,:),'r')
>> plot(ub(3,:),'y')
>> hold off



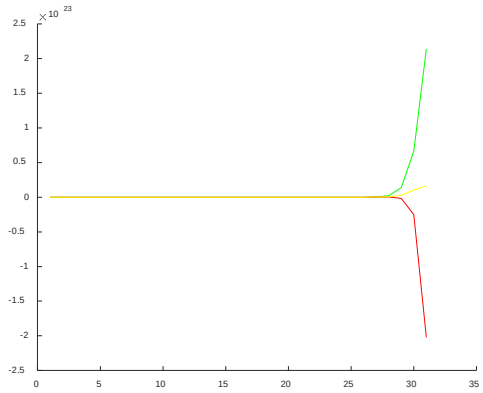
La suite converge toujours vers (10/3, 2/3, 10/3).

- Remplaçant A par A_b

```
>> Ab = [ 5 6 3 ; -1 5 -1 ; 1 2 0 ]
```

Pour un premier terme de la suite égale à u_{1b}, on a :

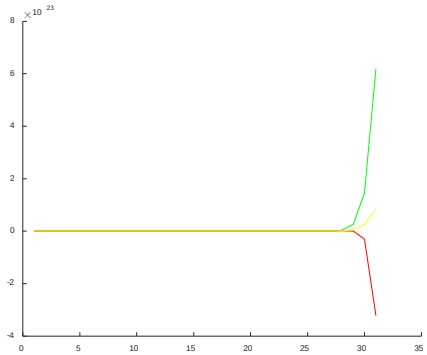
```
>> ub = suite(Ab,u1b,b,30);
>> hold on
>> plot(ub(1,:),'g')
>> plot(ub(2,:),'r')
>> plot(ub(3,:),'y')
>> hold off
```



La suite diverge !!

Pour un premier terme de la suite égale à u_1, on a :

```
>> ub = suite(Ab,u1,b,30);
>> hold on
>> plot(ub(1,:),'g')
>> plot(ub(2,:),'r')
>> plot(ub(3,:),'y')
>> hold off
```



La suite diverge quelque soit le premier terme de la suite.